

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
1 February 2001 (01.02.2001)

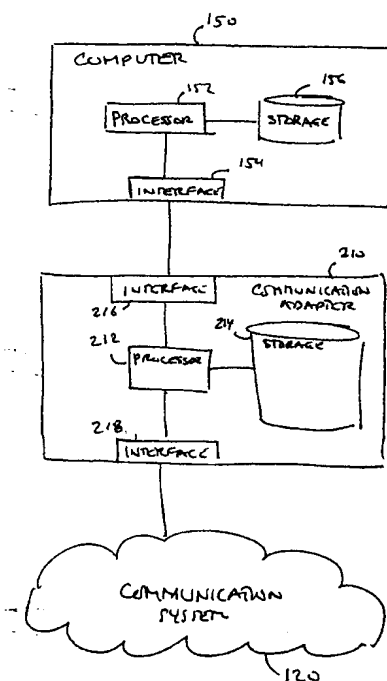
PCT

(10) International Publication Number  
**WO 01/07984 A2**

- (51) International Patent Classification<sup>7</sup>: **G06F** (74) Agent: **HOLMAN, John, Clarke**; Jacobson, Price, Holman & Stern, PLLC, The Jenifer Building, 400 Seventh Street, NW, Washington, DC 20004 (US).
- (21) International Application Number: PCT/US00/20180
- (22) International Filing Date: 26 July 2000 (26.07.2000) (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data: 60/145,736 27 July 1999 (27.07.1999) US
- (71) Applicant: **CAIS, INC.** [US/US]; 1255 22nd Street, N.W., Washington, DC 20037 (US).
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- (72) Inventor: **GOODMAN, David, D.**; 730 North Danville Street, Arlington, VA 22210 (US).

[Continued on next page]

(54) Title: UNIVERSAL SERIAL BUS ADAPTER WITH AUTOMATIC INSTALLATION



(57) Abstract: A peripheral device and a method for operating the peripheral device for automatic installation, in which the method includes coupling the peripheral device to a computer and sending a first device identification from the peripheral device to the computer. The peripheral device emulates a device of a type determined by the first device identification, including transferring a driver from the peripheral device to the computer. Then, the peripheral device sends a second device identification from the peripheral device to the computer, such that the sent device identification is for a device supported by the driver transferred to the computer. The peripheral device is then operated by interacting with the driver on the computer.



**Published:**

— Without international search report and to be republished upon receipt of that report.

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

## UNIVERSAL SERIAL BUS ADAPTER WITH AUTOMATIC INSTALLATION

### Cross-Reference to Related Applications

This application claims the benefit of U.S. Provisional Application  
5 No. 60/145,836 filed July 27, 1999, which is incorporated herein by reference.

### Background

This invention relates to an automatic installation of a communication adapter for a computer.

“Universal Serial Bus” (USB) refers to a particular standard that has been adopted  
10 by the electronics industry for establishing a communication between a computer and a number of peripheral devices. A full description of the USB standard can be found in “Universal Serial Bus Specification Revision 1.1” dated September 23, 1998 and available from the USB Implementers Forum, Inc. The USB provides 12 megabit per second digital communication over standardized five-conductor cables, which connect  
15 devices that are less than 20 feet apart, and also provides limited power connections between devices. One of the most important applications for USB communication is the connection ordinary personal computers, such as Microsoft Windows based personal computers, with peripheral equipment including input/output devices such as keyboards, and communication devices such as modems and Ethernet network interface adapters.

20 The USB standard not only dictates low-level communication characteristics, but also specifies how a USB-compliant device identifies itself to a host computer to which it is connected. In particular, when a USB device is attached to a computer, a startup protocol is defined in which the host computer receives information about the device.

For the host computer to make use of the functionality of the USB device, it  
25 generally requires software that is specific to that device, or at least specific to class of devices to which that device belongs and which share a common generic interface. This software is typically referred to as “driver” software. For example, if a pointing mouse connects to the USB port of a host computer, the host computer must understand that the data it inputs through the port must be used to control the movement of a pointer across  
30 the computer display. It must therefore relay this data, in the correct form, to the part of a governing program, typically an operating system, which controls these movements.

Referring to FIG. 1, when a USB device, such as a communication adapter 110, is first attached to an interface 154 on a computer 150, the operating system executing on

that computer typically determines whether it has available the appropriate driver software for that device in its storage **156** based on the identification of the device it receives over the USB connection. If it does not, the operating system may request that the user insert a removable storage **160**, such as a floppy disk, that has the driver so that it  
5 can be installed onto computer **150**. This procedure is typically used on the Microsoft Windows 98 operating system. Communication adapter **110** then provides communication services using the driver for communicating between computer **150** and other computers on a communication system **120**, such as the Internet.

Drivers for some devices are preloaded on certain operating system. For example,  
10 a generic keyboard driver, which supports keyboards from a variety of manufacturers, may already be loaded. If such a driver is loaded, when a USB keyboard is attached to the host computer, the user may not have to supply a specific driver for the keyboard if it is compatible with the generic driver.

#### Summary

15 In situations in which the operating system of a host computer does not have a suitable driver already available for a USB, using prior approaches, a user may be asked to insert a disk containing the driver so that it can be loaded. In certain situations, the user may not have such a disk. In a general aspect, the invention is a method for coupling a USB device, in particular coupling a USB device which provides an Ethernet interface,  
20 to a host computer without necessarily requiring that a driver for that USB device be already available to the computer or requiring that the user provide a disk with the driver. The method includes the USB device emulating another type of device, such as keyboard device, for which the host computer does have a suitable driver and passing a driver for the device over the USB interface to the computer while emulating the device. After the  
25 driver is loaded onto the computer, the devices makes use of the driver to operate in its native made rather than emulating another device.

In one aspect, in general, the invention is a method for operating a peripheral device. The method includes coupling the peripheral device to a computer and sending a first device identification from the peripheral device to the computer. The method then  
30 includes emulating a device of a type determined by the first device identification, including transferring a driver from the peripheral device to the computer. Then, the peripheral device sends a second device identification from the peripheral device to the computer, such that the sent device identification is for a device supported by the driver transferred to the computer. The peripheral device is then operate by interacting with the  
35 driver on the computer.

The invention can include one or more of the following features.

Sending the first device identification includes sending an identification of a keyboard device.

5 Transferring the driver to the computer includes emulating keystrokes for entering data on the computer.

Operating the peripheral device to interact with the driver includes providing communication services between the computer and a communication system.

Providing communication services includes passing Ethernet signals between the peripheral device and the communication system.

10 Coupling the peripheral device to the computer includes establishing communication between the peripheral device and the computer according to a Universal Serial Bus (USB) standard.

Sending the first device identification includes sending an identification of a storage device.

15 Transferring a file from the storage device to the computer.

In another aspect, in general, the invention is a peripheral device. The device has an interface for coupling the peripheral device to a computer, and a processor. The processor is programmed (a) to emulate a first type of device, (b) to transfer a driver for a second type of device through the interface to the computer while emulating the first type of device, and (c) to operate as the second type of device after transferring the driver to the computer.

The device can include one or more of the following features:

The interface includes a Universal Serial Bus interface.

The first type of device is a keyboard device.

25 The first type of device is a storage device.

The second type of device is a communication adapter.

The second type of device comprises an Ethernet communication adapter.

The invention has an advantage of neither requiring a specific device driver to be preloaded on the computer to use the USB device nor requiring that the user supply a disk with such a driver. For example, a hotel may equip each room with a cable to which guests can connect their laptop computers, and over which the computer can communicate over the Internet. A guest connects a USB device between his computer and the cable to provide provides an interface to the Internet. For example, if communication over the cable is according to the Ethernet standard, the USB device can be a USB Ethernet adapter that operates according to the present invention. According to the present invention, the guest does not have to manually install software on the

computer, a procedure that may be time consuming, prone to errors, or beyond the expertise of the guest.

#### Description of Drawings

FIG. 1 is a block diagram that illustrates a prior approach to configuring a peripheral device;

FIG. 2 is a block diagram that illustrates the present approach to configuring a peripheral device;

FIG. 3 is a flowchart of operation of a peripheral device; and

FIG. 4 illustrates multiple computers coupled through communication adapters to a communication system and to the Internet.

#### Description

Referring to FIG. 2, computer 150 has an interface 154 for connecting peripheral devices, such as communication adapter #0210, to the computer. In this embodiment, interface 154 is a USB interface. In alternative embodiments, the present approach can be used with different types of interfaces. Initially when communication adapter 210 is connected to interface 154, computer 150 does not necessarily have suitable drives in its storage 156 to operate communication adapter 210 in its native mode. Communication adapter 210, when operating in its native mode, provides a communication interface between computer 150 and a communication system 120. Communication adapter 210 is coupled to communication system 120 using an Ethernet standard, and communication system 120 includes a wiring network in a building that provides a communication link between computer 150 and the Internet. In alternative embodiments, communication adapter 210 does not necessarily use an Ethernet standard for communicating over communication system 120. In yet other alternative embodiments, other types of peripheral devices that are not communication adapters are configured using the present invention.

Communication adapter 210 includes an interface 216 for communicating with computer 150. Interface 216 is a USB interface. Communication adapter 210 also includes an interface 218 for communicating with communication system 120. Interface 218 is an Ethernet interface. Communication adapter 210 also includes a processor 212, such as a micro-controller, and a storage 214, such as a read-only semiconductor memory. Storage #0214 holds software instructions that are executed by processor 212. Generally, the instructions allow communication adapter 210 to operate in two operating modes. In a first operating mode, communication adapter 210 emulates a common

peripheral device for which a driver is already loaded in storage 156 on computer 150. In a second operating mode, a “native” mode, communication adapter 210 provides communication services that allow computer 150 to communicate over communication system 120. In this embodiment, in the first operating mode, communication adapter 210  
5 emulates a standard USB keyboard for which a driver is preloaded onto the Windows 98 operating system.

Referring to FIG. 3, when communication adapter 210 is initially connected to computer 150, the adapter makes use of the DC power that is available on the cable connecting to the computer. Processor 212 starts up emulating a known device. As part  
10 of that emulation, processor 212 sends data to computer 150 according to the USB standard that identifies the adapter as a USB-based keyboard (step 310). The data that it sends is stored in storage 214 either in conjunction with software instructions for controlling this first operating mode, or as a separate data block.

After computer 150 receives the data that identifies communication adapter 210  
15 as a keyboard, the computer communicates with communication adapter 210 as if the adapter were actually a keyboard.

After sending the initial data, processor 212 waits until enough time has passed for computer 150 to configure itself to communicate with the adapter. At that time, communication adapter 210 essentially has as much power to control computer 150 as an  
20 individual has seated at a keyboard. Communication adapter 212 uses these capabilities to upload a driver for its native operating mode on computer 150 (step 320).

Uploading of the driver (step 320) is performed as follows. Processor 212 first sends out a character code that would be sent by a keyboard if the “escape” key is pressed and released while the “control” key is held down. It then sends out a character code that  
25 would be sent if an “r” is pressed. This sequence of character codes invokes a “run” panel of Windows 98 which is being execute by processor 152 on computer 150. Under Windows 98, this effect does not depend on what application software computer 150 is currently executing.

After invoking the run panel, processor 212 sends out character codes that  
30 correspond to the keystroke sequence, “c:\windows\command.com” followed by the “enter” key. Computer 150 responds by opening up a “dos” window.

At this point, processor 212 sends the character codes that correspond to the keystroke sequence “debug” followed by “enter.” In response, processor 152 on  
35 computer 150 begins to execute the standard “debug” program, which is preloaded on storage 156. Using “debug,” it is generally possible for a computer operator to type in a data file of arbitrary length and content, store this file on the computer disk, and assign it

any particular name in the disk directory. As a result of communication adapter **210** emulating a keyboard, processor **212** also has the ability to upload a file.

Processor **212** then uploads a driver file by sending suitable character codes that correspond to inputting the contents of the driver file for processing by the “debug” program.

After placing the driver file on storage **156**, controller **212** communicates with interface **154** according to USB standard in such a manner that interface **154** acts as if a new device was connected to it. A USB interface provides a connection for multiple peripheral devices, and therefore does not require communication adapter to be physically disconnected and reattached. Processor **212** then signals to computer **150** that that a USB-Ethernet adapter has been connect (step **330**), and optionally, that indicate that the keyboard it was emulating has been disconnected. Processor **212** identifies itself as a type of communication adapter such that the operating system executing on computer **150** finds the driver that was previously uploaded by the communication adapter without requiring intervention by the operator.

Communication adapter **210** then operates in its native mode providing communication services between computer **150** and communication system **120**.

In an alternative embodiment, rather than uploading an entire driver file using the “debug” program, processor **212** uploads a bootstrap program, and then executes the bootstrap program on computer **150**. The bootstrap program, which executes on computer **150**, communicates with communication adapter **210** to retrieve the driver. In order to support communication between the bootstrap program executing on computer **150** and communication adapter **210**, communication adapter additionally emulates a USB-based storage device, such as a CD-ROM, and the bootstrap copies the driver file from the emulated CD-ROM to the computer’s storage. This approach may be faster than the first embodiment in which the entire driver is transferred using the “debug” program since it involves less overhead of emulating a keyboard. This approach relies on computer **150** including a suitable driver on its storage **156** for supporting an external USB-based CD-ROM drive.

In alternative embodiments, communication adapter **210** emulates other types of devices in order to upload a driver for its native mode. In one alternative embodiment, communication adapter **210** emulates a CD-ROM drive. A feature of the Windows 98 operating system is that a CD-ROM disk can include a program that is automatically invoked by the operating system when the disk is inserted into a CD-ROM drive. In particular, a file named “autorun.inf” provides information to the operating system that identifies a program file that is to be executed when the disk is inserted. In this



alternative embodiment, communication adapter **210** identifies itself as an external USB-based CD-ROM drive, in a manner that is similar to its identifying itself as a external USB-based keyboard in the previously described embodiments. Rather than emulating keystrokes, communication adapter **210** emulates a disk being inserted into the drive, and then emulates processing of requests from computer **150** to read the "autorun.inf" file and to read the program that is to be executed. This bootstrap program then transfers the driver to the computer's storage.

Other alternative types of devices may be emulated. If the emulated device has a provision for executing an externally-provided program on computer **150**, an approach of the type described above can be used.

The embodiments described above make use of the Microsoft Windows operating system. In alternative embodiments, communication adapter **210** includes software in storage **214** that is adapted to different and possibly multiple operating systems. Communication adapter **210** detects which operating system is used by computer **150** during the initial USB standard exchange, and performs suitable emulation and driver uploading procedures for that operating system.

In alternative embodiments, storage **214** is writeable, and is updated from a server over communication system **120**. In yet other embodiments, software is transferred over communication system **120** when it is needed and stored temporarily on storage **214**.

Referring to FIG. 4, an application of communication adapters **210** of the types described above is in a hotel communication system. A wiring network **410**, such as a twisted pair wiring network, which may be shared with ordinary telephone communication, links a hub **420** and each of a number of rooms. Each room includes a jack into which communication adapter **210** can be plugged. Hub **420** is coupled to a router **430** that provides a communication link to Internet **440**. In one embodiment, hub **420** and router **430** are located in a telephone wiring closet from which wiring network **410** runs to the hotel rooms.

In alternative embodiments, communication adapter **210** does not necessarily communicate with hub **420** using an Ethernet standard. Alternative signaling techniques may be used for passing data over one of two pairs of wires used for active telephone communication or for passing communication over other transmission paths, such as coaxial cable or using wireless transmissions.

It is to be understood that the foregoing description is intended to illustrate and not to limit the scope of the invention, which is defined by the scope of the appended claims. Other embodiments are within the scope of the following claims.

What is claimed is:

1. A method for operating a peripheral device (110) comprising:  
coupling the peripheral device to a computer (150);  
sending a first device identification from the peripheral device to the computer;  
5       emulating a device of a type determined by the first device identification, including  
transferring a driver from the peripheral device to the computer;  
      sending a second device identification from the peripheral device to the computer,  
wherein the send device identification is for a device supported by the driver transferred to the  
computer; and  
10       operating the peripheral device to interact with the driver on the computer.
2. The method of claim 1 wherein sending the first device identification includes  
sending an identification of a keyboard device.
3. The method of claim 2 wherein transferring the driver to the computer includes  
emulating keystrokes for entering data on the computer.
- 15       4. The method of claim 1 wherein operating the peripheral device to interact with the  
driver includes providing communication services between the computer and a communication  
system.
5. The method of claim 1 wherein providing communication services includes passing  
Ethernet signals between the peripheral device and the communication system.
- 20       6. The method of claim 1 wherein coupling the peripheral device to the computer  
includes establishing communication between the peripheral device and the computer according  
to a Universal Serial Bus (USB) standard.
7. The method of claim 1 wherein sending the first device identification includes  
sending an identification of a storage device.

8. The method of claim 1 includes transferring a file from the storage device to the computer.

9. A peripheral device (110) comprising:

an interface (216) for coupling the peripheral device to a computer; and

5 a processor (212) programmed (a) to emulate a first type of device, (b) to transfer a driver for a second type of device through the interface to the computer while emulating the first type of device, and (c) to operate as the second type of device after transferring the driver to the computer.

10 10. The device of claim 9 wherein the interface (includes a Universal Serial Bus interface.

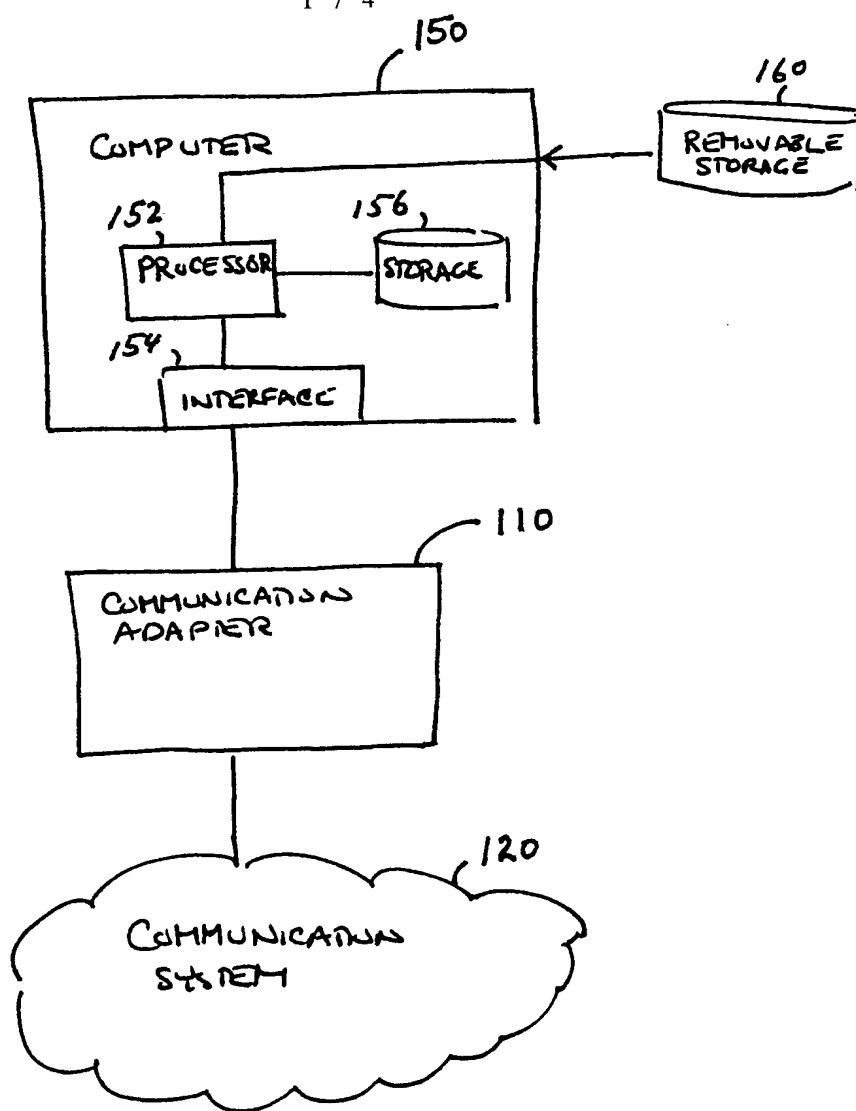
11. The device of claim 9 wherein the first type of device comprises a keyboard device.

12. The device of claim 9 wherein the first type of device comprises a storage device.

13. The device of claim 9 wherein the second type of device comprises a communication adapter.

15 14. The device of claim 1 wherein the second type of device comprises an Ethernet communication adapter.

1 / 4

FIG. 1 (PRIOR ART)

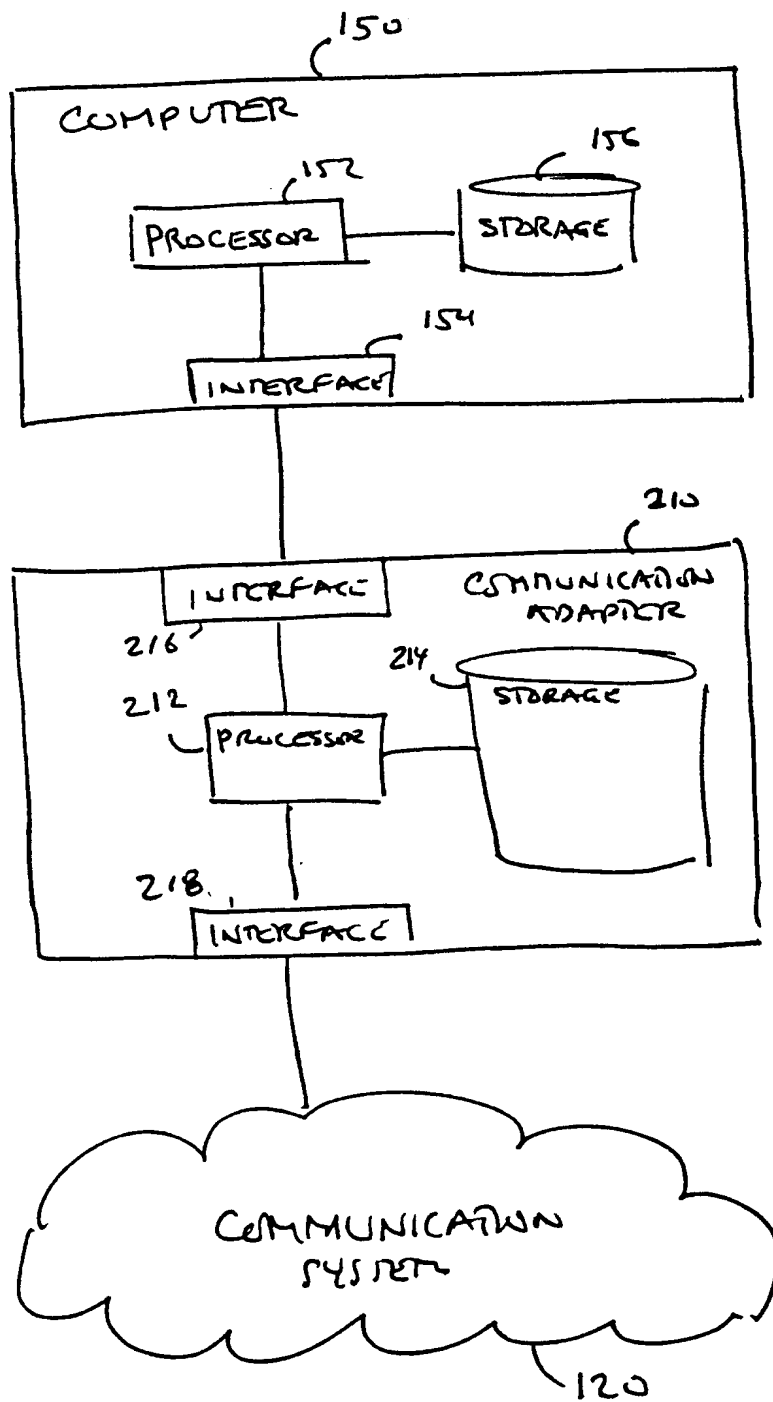


FIG 2 .

3 / 4

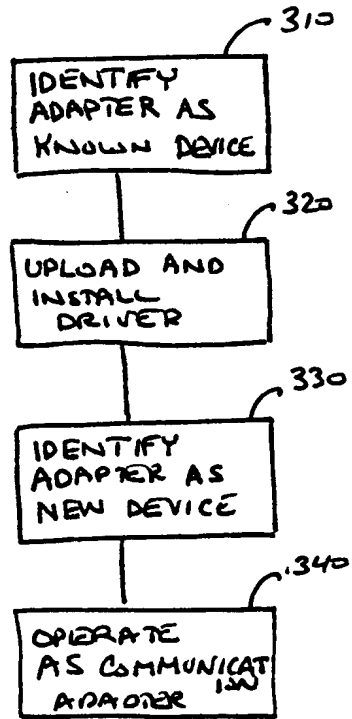


FIG.3

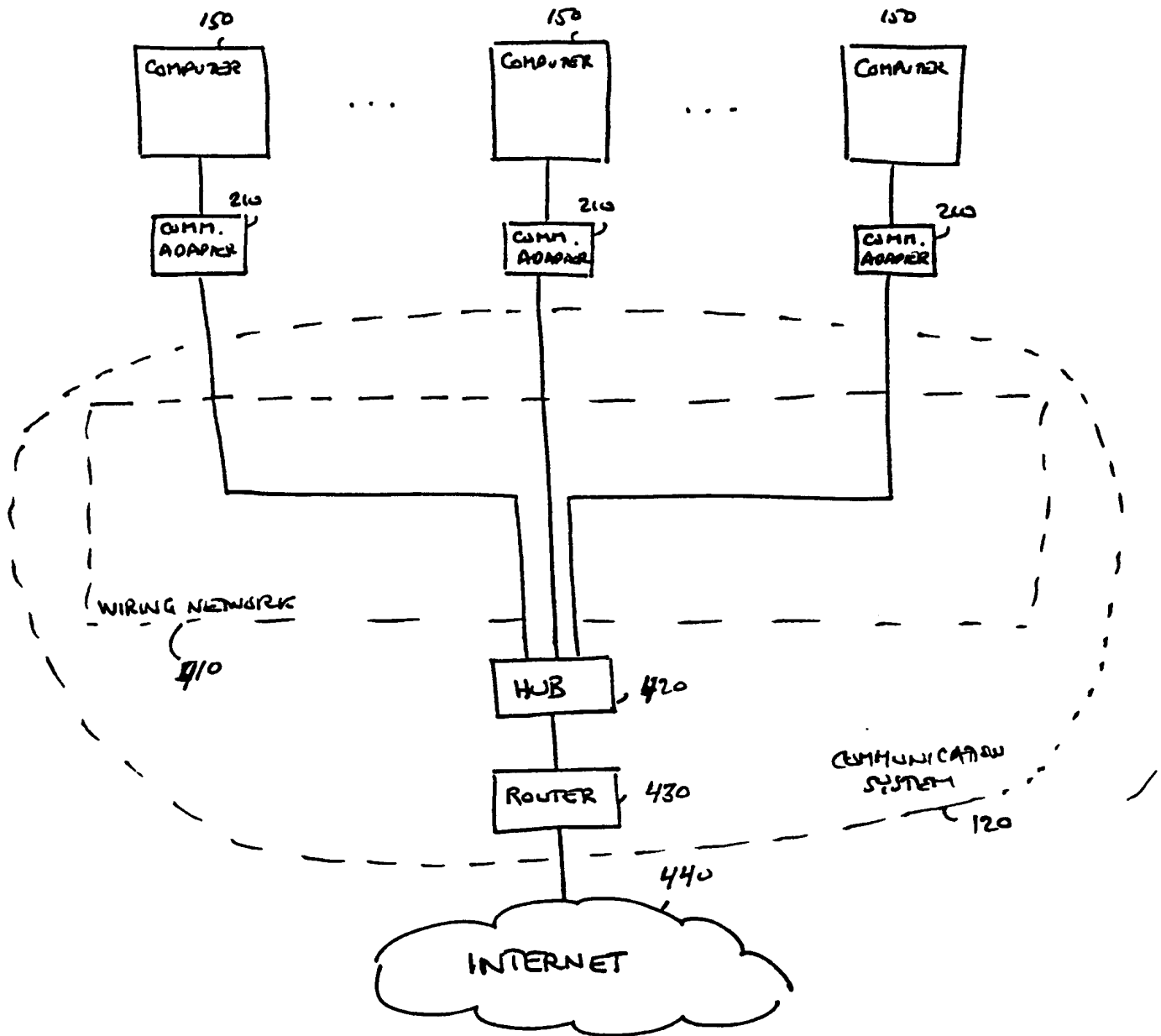


FIG. 4